

# Devoir 1

Martin Chaperot  
20205638

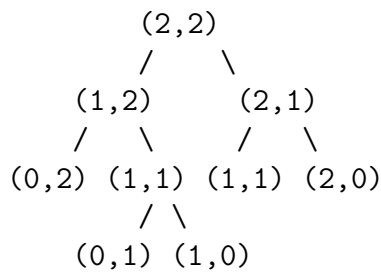
## 2

1

Temps:  $O(2^n)$

Espace:  $O(n)$

Cette fonction est une implementation récursive de la fonction d'Ackermann.



## 2

Temps:  $O(n^3)$

Espace:  $O(n^3)$

Cette fonction est une implémentation de la programmation dynamique pour résoudre le problème de la construction de la chaîne cible `target` en utilisant les sous-chaînes fournies dans la liste `pieces`.

## 3

Temps:  $O(n^2)$

Space:  $O(n^2)$

Cette fonction est une implémentation de la programmation dynamique pour déterminer s'il est possible de sommer des éléments du tableau `options` pour atteindre le nombre `target`.

## 4

La complexité temporelle augmenterait.

La complexité pour les deux fonctions est  $O(2^n)$ . Car pour chacune des fonctions, nous devons faire des calculs pour chaque element avec chaque autre element.

## 3.4

### 1

L'algorithme déplace une section à la fois de Nam à Pam, puis de Pam à Sam, en respectant l'ordre original. Ce processus se répète jusqu'à ce que toutes les sections soient déplacées de Nam à Sam.

Complexité temporelle:  $O(n)$

Complexité spatiale:  $O(n)$

### 2

Complexité temporelle:  $O(n)$

Complexité spatiale:  $O(n)$